```
normal_form_fusion  THM

⊢  ∀E:TaggedEventStruct, P,I,J,K:TraceProperty(E)
|  , R:(Trace(E)→Trace(E)→Prop).
|  tag_splitable(E;R) ⇒
|  (∀tr_1,tr_2:Trace(E). (tr_1 R tr_2) ⇒ (tr_2 R tr_1)) ⇒
|  R preserves P ⇒
|  R preserves K ⇒
|  (∀tr:Trace(E).
|   (I ∧ K)(tr) ⇒ (∃tr':Trace(E). I(tr') & J(tr') & (tr R tr')))
|  ⇒
|  (((I ∧ J) ∧ K) fuses P) ⇒ ((I ∧ K) fuses P)
| by Unfold `tag_splitable` 0

|    THEN Unfolds ``str_trace trace_property`` 0 THEN Auto

|    THEN Try (Unfold `trace_property` 0 THEN Auto)

|    THEN Try (Fold `trace_property` 0 THEN Auto)
|
1. E : TaggedEventStruct
2. P : (|E| List)→Prop
3. I : (|E| List)→Prop
4. J : (|E| List)→Prop
5. K : (|E| List)→Prop
6. R : (|E| List)→(|E| List)→Prop
7. ∀tr_1,tr_2:|E| List.
|  (tr_1 R tr_2) ⇒ (∀m:Label. <tr_1>_m R <tr_2>_m)
8. ∀tr_1,tr_2:|E| List. (tr_1 R tr_2) ⇒ (tr_2 R tr_1)
9. R preserves P
10. R preserves K
11. ∀tr:|E| List.
|   (I ∧ K)(tr) ⇒ (∃tr':|E| List. I(tr') & J(tr') & (tr R tr'))
12. ((I ∧ J) ∧ K) fuses P
⊢  (I ∧ K) fuses P
| by ParallelOp (-1) THEN All (Unfold `str_trace`)

|    THEN ParallelOp (-2) THEN D 0 THENA Auto

|    THEN ParallelOp (-2) THEN ExRepD
THEN InstHyp [tr'
|    ] (-8) THEN Auto
|
|
|\
| 1. E : TaggedEventStruct
| 2. P : (|E| List)→Prop
| 3. I : (|E| List)→Prop
| 4. J : (|E| List)→Prop
| 5. K : (|E| List)→Prop
| 6. R : (|E| List)→(|E| List)→Prop
| 7. ∀tr_1,tr_2:|E| List.
|    (tr_1 R tr_2) ⇒ (∀m:Label. <tr_1>_m R <tr_2>_m)
| 8. ∀tr_1,tr_2:|E| List. (tr_1 R tr_2) ⇒ (tr_2 R tr_1)
| 9. R preserves P
```

```
| 10. R preserves K
| 11. ∀tr:|E| List.
|     (I ∧ K)(tr) ⇒ (∃tr':|E| List. I(tr') & J(tr') & (tr R tr'))
| 12. ∀tr:|E| List.
|     (∀m:Label. P(<tr>_m)) ⇒ ((I ∧ J) ∧ K)(tr) ⇒ P(tr)
| 13. tr : |E| List
| 14. ∀m:Label. P(<tr>_m)
| 15. (I ∧ K)(tr)
| 16. tr' : |E| List
| 17. I(tr')
| 18. J(tr')
| 19. tr R tr'
| 20. m : Label
| ⊢  P(<tr'>_m)
|   by InstHyp [m] (-7) THENA Auto
| THEN AllHyps (\h. InstHyp [tr;tr';
|      m] h THENA Complete Auto)
|
|      ...
|
|\
| .....antecedent .....
| 1. E : TaggedEventStruct
| 2. P : (|E| List)→Prop
| 3. I : (|E| List)→Prop
| 4. J : (|E| List)→Prop
| 5. K : (|E| List)→Prop
| 6. R : (|E| List)→(|E| List)→Prop
| 7. ∀tr_1,tr_2:|E| List.
|    (tr_1 R tr_2) ⇒ (∀m:Label. <tr_1>_m R <tr_2>_m)
| 8. ∀tr_1,tr_2:|E| List. (tr_1 R tr_2) ⇒ (tr_2 R tr_1)
| 9. R preserves P
| 10. R preserves K
| 11. ∀tr:|E| List.
|     (I ∧ K)(tr) ⇒ (∃tr':|E| List. I(tr') & J(tr') & (tr R tr'))
| 12. ∀tr:|E| List.
|     (∀m:Label. P(<tr>_m)) ⇒ ((I ∧ J) ∧ K)(tr) ⇒ P(tr)
| 13. tr : |E| List
| 14. ∀m:Label. P(<tr>_m)
| 15. (I ∧ K)(tr)
| 16. tr' : |E| List
| 17. I(tr')
| 18. J(tr')
| 19. tr R tr'
| ⊢  ((I ∧ J) ∧ K)(tr')
|   by All (Unfold `prop_and`) THEN All Reduce THEN Auto
|
|      ...
|
.\
. 1. E : TaggedEventStruct
. 2. P : (|E| List)→Prop
. 3. I : (|E| List)→Prop
. 4. J : (|E| List)→Prop
. 5. K : (|E| List)→Prop
```

```
. 6. R : (|E| List)→(|E| List)→Prop
. 7. ∀tr_1,tr_2:|E| List.
. |  (tr_1 R tr_2) ⇒ (∀m:Label. <tr_1>_m R <tr_2>_m)
. 8. ∀tr_1,tr_2:|E| List. (tr_1 R tr_2) ⇒ (tr_2 R tr_1)
. 9. R preserves P
. 10. R preserves K
. 11. ∀tr:|E| List.
. |   (I ∧ K)(tr) ⇒ (∃tr':|E| List. I(tr') & J(tr') & (tr R tr'))
. 12. ∀tr:|E| List.
. |   (∀m:Label. P(<tr>_m)) ⇒ ((I ∧ J) ∧ K)(tr) ⇒ P(tr)
. 13. tr : |E| List
. 14. ∀m:Label. P(<tr>_m)
. 15. (I ∧ K)(tr)
. 16. tr' : |E| List
. 17. I(tr')
. 18. J(tr')
. 19. tr R tr'
. 20. P(tr')
. ⊢  P(tr)
. | by AllHyps(\h. Complete (
.
. |       Unfold `preserved_by` h THEN InstHyp [tr';tr
. |    ] h THEN Auto THEN UseSym
. |  ))
. |
. 1. E : TaggedEventStruct
. 2. P : (|E| List)→Prop
. 3. I : (|E| List)→Prop
. 4. J : (|E| List)→Prop
. 5. K : (|E| List)→Prop
. 6. R : (|E| List)→(|E| List)→Prop
. 7. ∀tr_1,tr_2:|E| List.
.    (tr_1 R tr_2) ⇒ (∀m:Label. <tr_1>_m R <tr_2>_m)
. 8. ∀tr_1,tr_2:|E| List. (tr_1 R tr_2) ⇒ (tr_2 R tr_1)
. 9. R preserves P
. 10. R preserves K
. 11. ∀tr:|E| List.
.     (I ∧ K)(tr) ⇒ (∃tr':|E| List. I(tr') & J(tr') & (tr R tr'))
. 12. ∀tr:|E| List.
.     (∀m:Label. P(<tr>_m)) ⇒ ((I ∧ J) ∧ K)(tr) ⇒ P(tr)
. 13. tr : |E| List
. 14. ∀m:Label. P(<tr>_m)
. 15. (I ∧ K)(tr)
. 16. tr' : |E| List
. 17. I(tr')
. 18. J(tr')
. 19. tr R tr'
. 20. P(tr')
. ⊢  P(tr)
.   by AllHyps(\h. Complete (
.
.         Unfold `preserved_by` h THEN InstHyp [tr';tr
.      ] h THEN Auto THEN EasyHyp
.    ))
END
```